④

AD-A221 479

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Aircraft Systems Technical Memorandum 125

A WORKED EXAMPLE OF JOB SIMULATION USING Micro SAINT (U)

by

C.M. Ford
J.G. Manton
and
P.K. Hughes

DTIC
ELECTE
MAY 14 1990
E
D
S

Approved for public release.

AUGUST 1989

90 05 14 046

AR-005-624

A WORKED EXAMPLE OF JOB SIMULATION USING Micro SAINT

by

C.M. Ford
J.G. Manton
and
P.K. Hughes

SUMMARY

Micro SAINT is a computer based simulation environment tailored to exercising models of human operators performing job activities. Models of human activity systems can be defined and exercised to determine the impact of various job design parameters on system performance. In this worked example, the activities of a ship's crew and the aircrew of an anti-submarine warfare helicopter are modelled during the period of preparation for a sortie. This model was used to gain experience with the Micro SAINT package and to explore some of the features of the package.

DSTO
MELBOURNE

CONTENTS

1. Introduction

A great deal of interest in aviation human factors is concerned with the cognitive, sensory and physical workload experienced by aircrew. This interest is partly due to recent changes in aircraft systems and cockpit display technology in which single purpose instruments are being replaced by multifunction and dedicated electronic displays that present integrated and (supposedly) simplified graphical information (Hart, 1988). With the rapid expansion of aircraft system capability due to advanced technologies, it is important that aircrew can fully utilize the functions provided. Inability to do so because of excessive workload demands will limit the operational performance of the aircraft.

The most frequently used methods of investigating operator workload are based on empirical studies and therefore depend on the availability of human operators and a completed system, or at least a prototype of the system (Hills et al., 1987). On the other hand, analytic methods are used to predict workload at an earlier stage of system design without requiring operators or systems. SAINT (Systems Analysis of an Integrated Network of Tasks), an example of an analytic technique, is a workload simulation computer program which attempts to predict human performance using a task network modelling framework. Operator or system performance is represented in a task network by attributes of tasks such as the mean, standard deviation and shape of the distribution of task completion times. Relationships between tasks are represented by branching and looping logic that determines the sequence of tasks in the network. The probabilistic nature of the branching logic and the variable completion times of individual tasks means that repeated execution of the model will generate many different paths through the network and consequently a range of completion times for the whole network. Exercising the model can therefore indicate how a system with a human operator might perform.

SAINT was developed at the Armstrong Aerospace Medical Research Laboratory of the USAF for use in human factors studies with P-GERT networks and Siegal and Wolf man-machine models (Chubb, 1981). SAINT is run on a mainframe computer and requires some familiarity with Fortran programming (Laughery and Drews, 1985). Micro SAINT was developed for IBM personal computers in order to overcome some of the cumbersome language problems encountered with SAINT. The personal computer version incorporates a more easily learnt interface but there is some loss of flexibility and power compared with the mainframe version. Micro SAINT has an interactive user environment for specifying and developing task network models and therefore permits development of task networks in a relatively straightforward manner.

The initial use of the Micro SAINT modelling environment was to predict the time aircrew take to complete sequences of tasks in the Royal Australian Navy's Seahawk helicopter. The Seahawk helicopter is due to come into service in 1990 but will initially go through an operational test and evaluation (OT & E) assessment. The predictions of the model will be compared with performance of aircrew during the OT & E phase. If the model can be validated then it will become a powerful tool for assessing changes in aircrew workload when additional weapon and sensor systems are proposed.

The first stage of the model building process is task analysis in which a particular job is described in terms of its constituent elements. For demonstration purposes we have used an analysis performed by the Naval Work Study Team who interviewed Fleet Air Arm personnel with experience of S-2 Tracker operations (the S-2 Tracker and the Seahawk have many functional characteristics in common). The interviewers examined the procedures that might be followed by the crew of a Seahawk helicopter in preparing for an anti-submarine warfare (ASW) sortie flown from a ship. This task analysis identified the tasks that each member of the crew would perform, the estimated time to complete each task and whether aircrew would interact with another crew member to achieve a particular task goal. From this

information a network diagram was developed. The network diagram "Testseahawk" in Figure ' gives an overview of the tasks performed by each member of the aircrew and ship's personnel.

This paper describes the requirements for constructing a network and generating valid results from a task network simulation.

2. Building a Task Network Model

There are two aspects to building a task network: firstly designing the network, and secondly, entering the network details into the Micro SAINT application (Laughery, 1985). The task analysis process is used to identify all the tasks to be performed by the operators, the order in which they are performed and any interactions between tasks.

In the Micro SAINT application a network represents the compilation of all tasks performed and includes their order and any interactions between tasks, whereas a task is a description of an event or activity. Networks and tasks can represent the activity of a machine or system, or the activities of a human operator. Within the Micro SAINT network structure each task is represented by a node. Each node consists of attributes such as identification number, name, time to complete, the names of following tasks and any user defined relationships to describe the effect of task completion on the system.

The operator interface to Micro SAINT is a series of menus and all stages of the program appear in the main menu. The interactive simulation environment is presented in four discrete sub-programs: *model development, model execution, analysis of results* and *utilities*.

The *model development* sub-program allows the model developer to enter details of all tasks and the conditions under which the model is required to operate. The *modify task menu* is used to enter information such as task number, name, mean completion time and standard deviation, any following task(s) and any user defined relationships. The user defined relationship attributes include the *time distribution type* and *task release conditions*. The *time distribution type* specifies the characteristics of the statistical distribution as either normal, exponential, gamma or rectangular and also the distribution's mean and standard deviation. The *task release condition fields* specifies the conditions under which a task will be 'released' (release in Micro SAINT terminology is a condition that allows a task in the network to start). The *decision type field* specifies the following task(s) or network(s) and their relative probabilities of occurrence. The five decision type alternatives are:

(a) *last task*, meaning no more tasks or networks will execute as a result of the completion of this task

(b) *single choice*, where there is only one possible following task or network

(c) *probabilistic*, where one of several alternative tasks or networks will follow. The probability of each task occurring is determined by a probability expression

(d) *tactical*, where one of several alternative tasks or networks will follow. The possible occurrence of each task is determined by an expression consisting of numeric constants, variables and mathematical or logical operators which calculates to a single value. The expression that calculates to the highest value indicates the next task or network to execute

(e) *multiple*, where there are several simultaneous tasks or networks to follow.

There are other features of model development which permit manipulation of the network to satisfy the modelling problem. The *job queue field* facility specifies which job executes next when there are multiple jobs travelling through a network. This facility creates a job queue

3

Figure 1 : Network diagram for Model "Testseahawk".

and tasks wait to be released through the network until other jobs have been completed. The *variable catalog* (sic) is a table where all the variables relevant to the modelling problem are specified. Variables can represent system or control attributes. There is also a facility to generate *arrays* if the same variable name is used to index an ordered set of variables. The *function library* permits development of functions which may be needed several times during a model. The *continuous variable* feature allows changes of variables in the model at regularly scheduled intervals and may permit modelling of factors such as fatigue or stress. The *simulation scenario* feature can be used to specify a change to a variable in the model at specified clock times. The *snapshots* facility extracts information about model variables at defined times during model execution. Snapshots can be created at specified times on the clock, at particular job queues or at specified tasks.

Model execution is the part of the program that permits the model to run. The *display mode menu* has the following five alternatives: *reticent, medium, verbose, event queue and debug*. The *debug mode* is used during the initial stages of model development and execution because it provides the greatest amount of information about the model's structure and indicates any errors that occur during execution of the model. *Reticent* may be used after the model has been debugged and is being run for data collection. The other three modes provide different amounts of information which may be useful if the effect of various variables, job queues or simulation scenarios is required. The model execution menu has a facility to turn the snapshots or execution trace on or off depending on whether the model is executed to collect data or not. A different *random number seed* is generated by Micro SAINT for each execution of the model and is used to calculate task completion times and probabilistic paths through the network. When *standard deviation* equals zero, the model and all individual tasks execute for their average time compared to their various execution times if the standard deviations of task completion times are non-zero. The *variables to display* option permits the values of the selected variables to be displayed on the screen during model execution. The values of variables can be reset by entering expressions in the *manipulate variables* field. The *number of times to execute* defines how many times the model iterates. Siegal and Wolf (1969) suggested that between 100 and 200 runs of a model will generate stable results.

The analysis of results section of the program displays the results from any model execution. After each iteration of the model, data are stored in four different results files. The *error files* (.ERR) stores the list of error messages of the most recent execution of the model. *Snapshot results files* (.SSR) stores the values of variables collected from snapshots. The *travel time files* (.TTM) stores all the run times. The *execution trace* (.TRC) stores the value of the system clock at the beginning and the end of each executed task. One of the options in the *execution trace* is a time line graph which shows the duration of different tasks.

The *utilities* sub-program *copies, deletes* or *merges* the model file(s). It also has a facility to import data from an ASCII file.

Additional information about Micro SAINT is contained in the Micro SAINT User's Guide (Archer, Drews, Laughery and Dahl, 1987).

3. The Seahawk Models

The aim of the present simulation was to assess the workload of all seven members of the helicopter and shipboard crew in preparing a Seahawk helicopter for an ASW sortie. Figure 1 shows the model "Testseahawk" in which the jobs of each of the seven crew members are grouped under different sub-networks which are in turn all grouped under network 0 which is the network representing the closed system of preparing for the sortie.

The seven sub-networks are:

    (a) FDMT (flight deck maintenance team - non-aircrew)
    (b) Pilot (aircrew)
    (c) Senso (sensor operator - aircrew)
    (d) Tacco (tactical coordinator - aircrew)
    (e) PWO (principal warfare officer - ship's crew)
    (f) ASAC (anti-submarine aircraft controller - ship's crew)
    (g) OOW (officer of the watch - ship's crew).

The sub-network for each member of the team consists of all the jobs they perform. For example the PWO's jobs include :

    (1) read OPGENS (operational planning documents)
    (2) read the flying brief details
    (3) prepare the flying brief
    (4) attend the briefing.

These jobs are represented in a network diagram in Figure 2.
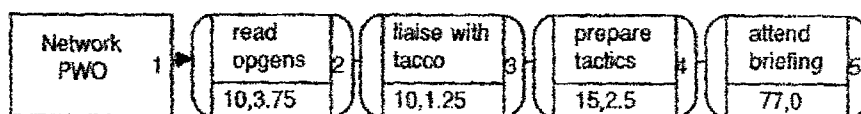

Figure 2. Network diagram for the PWO. The number at the right of each node identifies a specific network (rectangle) or task (rounded-corner rectangle). The title of the task or network is displayed in the upper half of each node and the mean and standard deviation (min) of task completion times are displayed in the lower half.

During the task analysis stage of model development it was realized that completion of some tasks required the presence of two crew members. Therefore those tasks common to two people had to occur simultaneously in the relevant sub-networks. This was accomplished by creating variables in the *variable catalog* which were then used in the *release condition* and *task ending effect* of particular tasks.

For example, task F4 and task S9 in Figure 1 are the tasks in which the FDMT and the Senso liaise with one another. For this to occur in the model, F3 and S8 each had in their *task ending effect* a variable which changed from 0 to 1 at the completion of the task. Tasks F4 and S9 had a *release condition* which required both the variables in the *ending effect* of F3 and S8 to equal 1. Tasks F4 and S9 will then be released simultaneously after the completion of F3 and S8.

It was also apparent during the task analysis stage of model development that the number of jobs performed by each person in the network was not uniform and that some individual crew members had a greater influence on the amount of time taken to complete the whole network compared to other crew members. This also meant that some of the crew encountered periods of time unallocated to any task in the network. However, in a real system, this time may be allocated to other activities not identified in the initial task analysis. The unallocated time between two successive tasks was determined by variables which flagged the end of one task and the beginning of the next task.

For example, the FDMT can complete their first two tasks independently of anyone else but their third task is to liaise with the Senso. However the FDMT cannot perform this task until the Senso has completed tasks S2 to S8. Therefore the FDMT has unallocated time between tasks F3 and F4. The variable fe reads the clock when task F3 had been completed, ft reads the clock once F4 had started and $ftime$ (unallocated time) was calculated from $fe-ft$.

The unallocated times of each crew member were collected during model execution and the data stored in snapshot files. Snapshot files were also created to accumulate the time each member was busy with assigned tasks.

After the network structure has been defined in *model development* the model can be run using *model execution*. The model was initially run in the debug mode several times so that errors could be easily identified and rectified. The model was then run to collect data and for the purposes of the present study it was executed 100 times with a different seed on each iteration. A random number generator in Micro SAINT is used to provide the seed.

4. Results

The unallocated and busy time results from the model "Testseahawk" indicated that the pilot, FDMT and PWO experienced the longest unallocated times, probably because all the networks started at the same time. However, in a real system, it is unlikely that all crew members will start their jobs at the same time. A second model was therefore developed which incorporated the tasks performed by the pilot, FDMT and PWO at times which would be expected to reduce their total unallocated time. Figure 3 is a network diagram of the new model called "Tactical" and shows the staggered start times of the pilot, FDMT and PWO. It should be noted that the changes incorporated in the new model reflect only one possible solution to the problem of reducing the unallocated time in the model and are presented here only to demonstrate the flexibility available in network modelling with Micro SAINT.

Figures 4 and 5 show the unallocated times from the two models and Figures 6 and 7 show the busy times for each model. The modifications incorporated in the "Tactical" model reduced the average unallocated time of the FDMT, pilot and PWO by 45.4 min, 26.0 min and 24.0 min respectively. Each average unallocated time of the Senso, Tacco and ASAC increased by only 3 min.These increased times were due to waiting periods for other tasks to be completed in the network by the crew members who started later.

A stepwise regression of clock time as a function of the mean unallocated times of each crew member was conducted separately for each model. The resultant equation for the "Testseahawk" model included terms for the unallocated times of the FDMT, Pilot, Tacco, Senso and PWO. The equation explained 92.0% of the variance of clock time and the ANOVA for the model was significant, $F(5,99)=228.72$, $p<0.01$. The equation for the "Tactical" model data included the above terms except for FDMT and Pilot, and accounted for 72.0% of clock time variance, $F(4,99)=62.2$, $p<0.01$.

Figures 6 and 7 show that the modifications incorporated in the second model did not have an effect on busy times for any crew member. This is an expected result because the number of tasks and their average completion times were the same for each model.

Average clock times for the two models in Figures 6 and 7 are approximately equal. This is probably because the busy times of the Tacco and the Senso had the greatest effect on total system time in both cases, as is evident from the high correlations of 0.79 between clock time and Tacco busy time in the "Testseahawk" model (Table 1) and 0.80 for the "Tactical" model (Table 2). The next highest correlations between clock time and busy time are for the Senso.
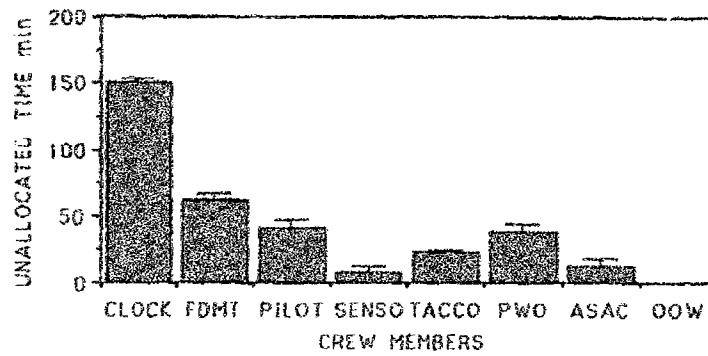
Figure 3 : Network diagram for model "Tactical".

Figure 4. Means and standard deviations of unallocated times for model "Testseahawk".
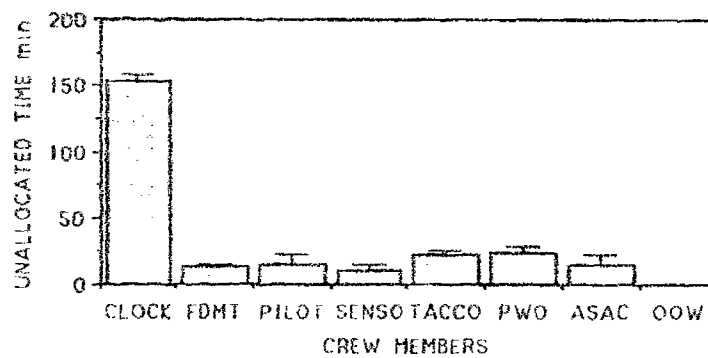Clock time is the mean duration of the whole network.

Figure 5. Means and standard deviations of unallocated times for model "Tactical".
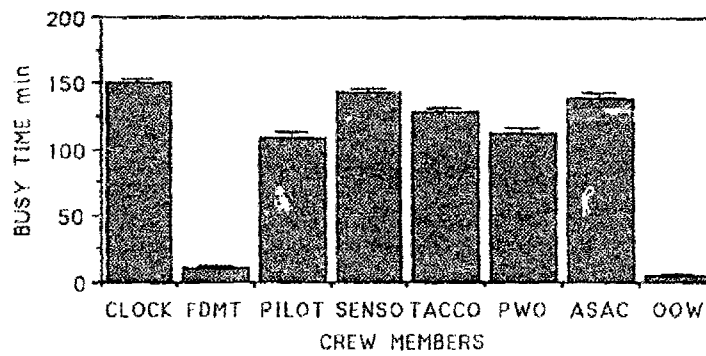Clock time is the mean duration of the whole network.

Figure 6. Mean busy times and standard deviations for model "Testseahawk". Clock time is the mean duration of the whole network.
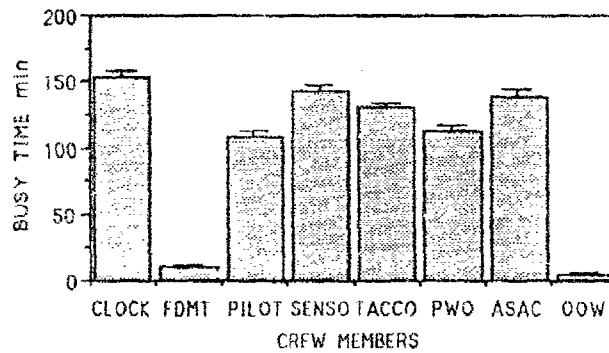

Figu. ⁻ Mean busy times and standard deviations for model "Tactical" Clock time is the mean duration of the whole network.

Table 1: Pearson correlation coefficients between clock time and busy times of the seven crew members in the model "Testseahawk".

	CLOCK	FDMT	PILOT	SENSO	TACCO	PWO	ASAC
FDMT	-0.164						
PILOT	-0.28	0.123					
SENSO	0.333	-0.029	-0.116				
TACCO	0.794	-0.154	-0.315	0.056			
PWO	0.01	-0.072	0.065	-0.185	-0.004		
ASAC	-0.154	-0.01	0.107	0.155	-0.136	-0.065	
OOW	-0.155	-0.122	0.15	-0.058	-0.126	0.013	-0.03

Table 2. Pearson correlations between clock time and busy times of the seven crew members in the model "Tactical".

	CLOCK	FDMT	PILOT	SENSO	TACCO	PWO	ASAC
FDMT	0.026						
PILOT	-0.094	-0.044					
SENSO	0.338	0.049	-0.009				
TACCO	0.801	0.006	-0.087	0.033			
PWO	0.266	-0.113	0.001	-0.091	0.334		
ASAC	-0.102	0.008	0.063	-0.02	-0.171	-0.098	
OOW	-0.127	-0.132	0.12	-0.032	-0.138	0.074	-0.068

5. Discussion

The Micro SAINT program has certain limitations which may affect the execution of large network models. Although a network may comprise up to 400 jobs (either sub-networks or tasks) the job which executes first in the network must be specified. For example, the seven sub-networks in the "Testseahawk" model could not begin simultaneously at the start of the network. Instead, the upper network 0 contained a task *start* which had a mean duration of zero and standard deviation of zero. Its *following tasks* statement was assigned *multiple* so that more than one task or network could begin simultaneously after *start's* execution. Zero duration tasks in the network can be also used as a 'trigger' for the snapshot files to collect data. (See task *end* in Figures 1 and 3.)

Another restriction of Micro SAINT is the limit of seven following tasks that can be entered into the *modify task menu*. This can be a problem in large networks where there are many alternative paths through the model.

At this stage the models Testseahawk and Tactical have no implications for actual operations as there was insufficient detailed data to derive a comprehensive task network. The exercise reported in this paper was to gain knowledge of the task description requirements and operation of the Micro SAINT application. It is planned that Micro SAINT will be used to model other phases of an ASW scenario when data on task completion times have been

collected. These data can be collected either in an operational vehicle or a simulator and will be used to develop a model of aircrew tasks or to validate a model constructed at an earlier stage of system development. Micro SAINT can also be used to predict the effects of modified hardware, software and operator task procedures on total system performance.

6. Conclusion

Micro SAINT is an important analytic tool that demonstrates how job and task design variables affect system performance. The simulation technique provides a method for analyzing operator workload and can indicate the effect of operator performance on system capability. Micro SAINT is a useful design and research tool for demonstrating and investigating the impact of possible modifications of aircrew-system interfaces.

References

Archer, R. A., Drews, C., Laughery, K. R. and Dahl, S.G. (1987). Micro SAINT User's Guide. Boulder, Colorado: Micro Analysis and Design.

Chubb, G. P. (1981). SAINT, a digital simulation language for the study of manned systems. In L. Moraal and K. -F. Kraiss (Eds), Manned Systems Design. New York: Plenum Press.

Laughery, K.R. and Drews, C. (1985). Micro SAINT: A computer simulation system designed for human factors engineers. In Proceedings of the Human Factors Society 29th Annual Meeting (pp 1061-1064). Santa Monica, California: Human Factors Society.

Laughery, K.R. (1985). Modeling human operators on a microcomputer. A micro version of SAINT. Simulation, 44, 10-16.

Hart, S. (1988). Helicopter human factors. In E. L. Weiner and D. C. Nagel (Eds), Human Factors in Aviation. New York: Academic Press Inc.

Hill, S. G., Plamondon, B. D., Wierwille, W. W., Lysaght, R. J., Dick, A. O. and Bittner, A. C. (1987). Analytic techniques for the assessment of operator workload. In Proceedings of the Human Factors Society 31st Annual Meeting (pp 368-372). Santa Monica, California: Human Factors Society.

Siegal, A. I. and Wolf, J. J. (1969). Man-Machine Simulation Models. Psychosocial and Performance Interaction. New York: John Wiley and Sons, Inc.

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central
Chief Defence Scientist
AS, Science Corporate Management (shared copy)
FAS, Science Policy (shared copy)
Director, Departmental Publications
Counsellor, Defence Science, London (Doc Data Sheet Only)
Counsellor, Defence Science, Washington (Doc Data Sheet Only)
S.A. to Thailand MRD (Doc Data Sheet Only)
S.A. to the DRC (Kuala Lumpur) (Doc Data Sheet Only)
OIC TRS, Defence Central Library
Document Exchange Centre, DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne
Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratory
Director
Library
Chief - Aircraft Systems Division
Authors: J. G Manton
             C.M. Ford
             P.K. Hughes

Materials Research Laboratory
Director/Library

Defence Science & Technology Organisation - Salisbury
Library

Navy Office
Navy Scientific Adviser

Army Office
Scientific Adviser - Army (Doc Data sheet only)

Air Force Office
Air Force Scientific Adviser (Doc Data sheet only)
CO Institute of Aviation Medicine, Point Cook (2 copies)

Department of Transport & Communication
Library

SPARES (20 COPIES)
TOTAL (59 COPIES)

AL 149
DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

1a. AR NUMBER AR-005-624	1b. ESTABLISHMENT NUMBER ARL-SYS-TM-125	2. DOCUMENT DATE AUGUST 1989	3. TASK NUMBER NAV 88/048

4. TITLE A WORKED EXAMPLE OF JOB SIMULATION USING Micro SAINT	5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF. (C)	6. NO. PAGES 13

RESTRICTED (R), UNCLASSIFIED (U)).

U	U	U
DOCUMENT	TITLE	ABSTRACT

7. NO. REFS.

7

| 8. AUTHOR(S) C.M. Ford J.G. Manton P.K. Hughes | 9. DOWNGRADING/DELIMITING INSTRUCTIONS

Not applicable |
|---|---|

10. CORPORATE AUTHOR AND ADDRESS **AERONAUTICAL RESEARCH LABORATORY** **P.O. BOX 4331, MELBOURNE VIC 3001**	11. OFFICE/POSITION RESPONSIBLE FOR: SPONSOR _____ DNAW _____ SECURITY _____ DOWNGRADING _____ APPROVAL _____ CSYD _____

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT)

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DSTIC, ADMINISTRATIVE SERVICES BRANCH, DEPARTMENT OF DEFENCE, ANZAC PARK WEST OFFICES, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO . . . .

No limitations

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE   [X] UNRESTRICTED OR   [ ] AS FOR 13a.

14. DESCRIPTORS Human Factors Engineering Computerized Simulation Environment Simulation	15. DISCAT SUBJECT CATEGORIES 0095D 0062F

16. ABSTRACT
Micro SAINT is a computer based simulation environment tailored to exercising models of human operators performing job activities. Models of human activity systems can be defined and exercised to determine the impact of various job design parameters on system performance. In this worked example, the activities of a ship's crew and the aircrew of an anti-submarine warfare helicopter are modelled during the period of preparation for a sortie. This model was used to gain experience with the Micro SAINT package and to explore some of the features of the package.

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT).

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER	19. COST CODE	20. TYPE OF REPORT AND PERIOD COVERED
Aircraft Systems Technical Memorandum 125	73 7020	Progress Report to June 1989

21. COMPUTER PROGRAMS USED

Micro-SAINT statview

22. ESTABLISHMENT FILE REF.(S)

M2/913

23. ADDITIONAL INFORMATION (AS REQUIRED)